APPLICATION NOTE

Atmel AT01244: DALI Slave Reference Design

Atmel 8-bit Microcontrollers

Features

- Compliant with international standard IEC62386-101, 102, 207
- Reference firmware design includes application, service and drivers
- Reference hardware made of three main parts:
  - DALI interface, which bridges the voltage between DALI bus and microcontroller
  - Microcontroller, which communicates with DALI master and controls the LEDs
  - LED driver, which drives the LED controlled by microcontroller

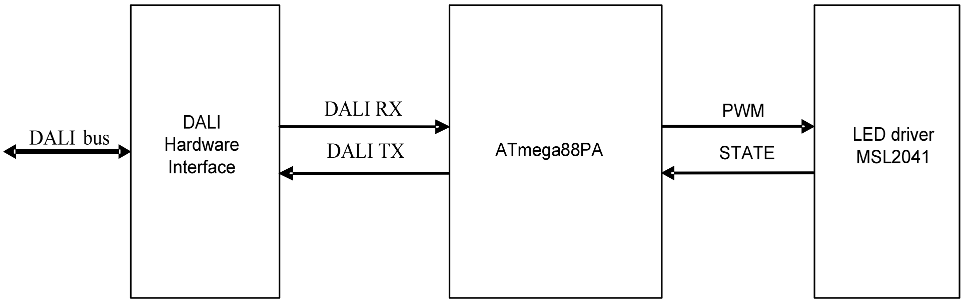
Description

DALI (Digital Addressable Lighting Interface) is an international standard created for lighting control and now becomes one of the more popular lighting interfaces today. DALI protocol is a digital communication interface with master-slave structure, aiming as a control gear between power supply and LED.

DALI slave communicates with DALI master based on half duplex signal with baud rate of 1200 bit/s. Frame is defined as the sequence of bits used to transmit data. Forward frame is from master to slave and backward from slave to master. The definition of bit and frame can be found in IEC62386-102.

Figure 1 shows the DALI slave system block diagram based on Atmel® megaAVR® device.

Figure 1. DALI slave system block diagram.



For this reference design, the hardware design files (schematic, BoM and PCB gerber) and software source code can be downloaded from Atmel website. The provided hardware documentation can be used with not limitations to manufacture the reference hardware solution for the design.

Table of Contents

1. Related Items

The following list contains links to the most relevant documents, software, and tools for DALI slave:

- IEC 62386-101, IEC 62386-102, IEC 62386-207
  The IEC standards define the specific of DALI slave.
- Atmel ATmega88PA datasheet
  ATmega88PA is the microcontroller used in this solution.
- Atmel LED Drivers-MSL2041/MSL2042 datasheet
  MSL2041 is used as LED driver.

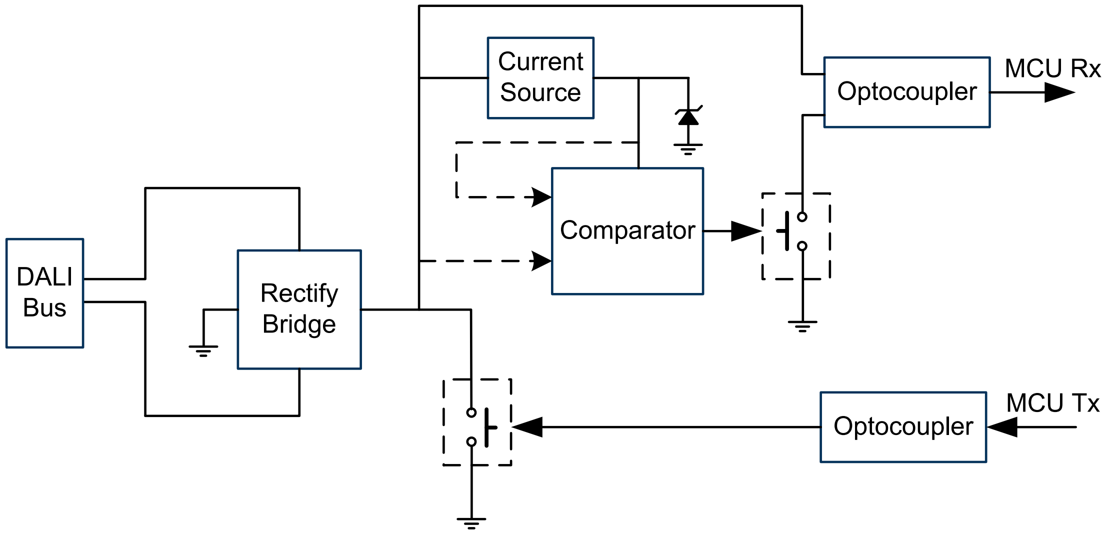
2. Hardware Modules

As previously described, there are mainly three hardware modules.

2.1 DALI hardware interface

DALI hardware interface bridges the voltage level between DALI bus and microcontroller. According to the DALI protocol, the HIGH level is 9.5V to 22.5V, and the LOW level is -6.5V to 6.5V. The special logic level should be converted to the logic which MCU can accept. The interface logic should be 0 to 5V for the MCU (Atmel ATmega88PA) used in this design.

Figure 2-1. DALI hardware interface.



Two optocouplers are used to isolate the voltage level between the DALI bus and microcontroller.

Using the comparator in above circuit has advantage over other simplified design in:

- Suitable for all working temperature range
- Trim the input waveform of signal for MCU
- Tolerant the difference of components for mass production

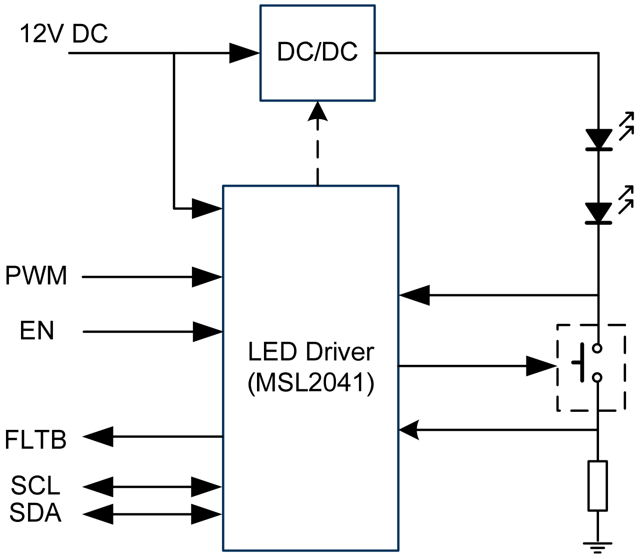
2.2 Microcontroller

ATmega88PA is one of the most popular devices in the Atmel MCU family. It is used as the main microcontroller in this reference design. ATmega88PA handles the commands and data which comes from the DALI interface part.

2.3 LED driver

Atmel MSL2041 is used as the LED driver in this design. It drives and controls the LEDs based on the input PWM signal from MCU. It also monitors the current and voltage of LEDs and feeds back to MCU the error status if error occurs.

Figure 2-2. DALI LED driver.

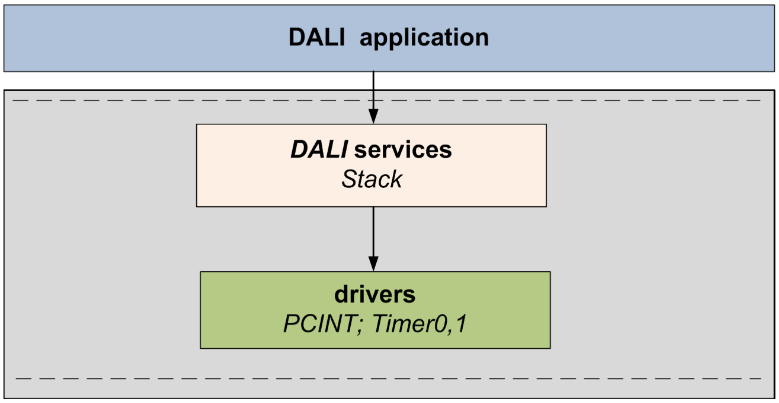


3. Firmware Modules

The firmware is developed based on layer structure as Atmel ASF that makes it more convenient to port between different Atmel device platforms. From bottom to top, the structure includes:

- **Drivers**
  Drivers are used to transmit DALI bits between top layer and afford timer.
  Timer0 is defined for DALI encode/decode and DALI frame timing.
  PCINT is used for DALI decode.
  Timer1 is used for fade timing and other system timing.
- **Service**
  Service processes frame sequence timing and implement DALI commands. To make migration feasible between different MCU device series, the service should be hardware irrelevant.
- **Application**
  Application realizes the DALI LED slave function. Besides DALI service, it also process power on, interface state detection, etc.

Figure 3-1. DALI firmware structure.



3.1 Drivers

The driver layer aims at timer and encode/decode of DALI bits. For Atmel Atmega88PA in this solution, PCINT and timer0/1 are chosen as the driver components. It is free to use other components if different method is used. PCINT detects the logic change in the input signal pin after initialization. The signal change triggers the PCINT interrupt and makes ISR handle the decode process.

Timer0 is configured as the time base for DALI encode/decode. It needs to count fast enough to measure the 1200 bit/s bit rate in DALI data stream. Timer0 needs to work together with PCINT to decode DALI signal. Timer1 is mainly used as time base for LED power output dimming.

3.2 Service

3.2.1 Frame State Convert

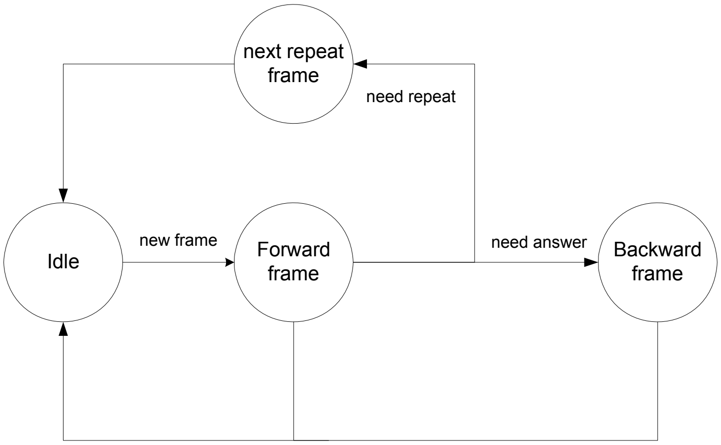
Firstly DALI slave goes into idle state and does state initialization. After this, the slave shall go to forward frame state immediately and wait for receiving DALI frame.

If received, DALI frame should be processed. According to different conditions, it goes to next repeat state, backward state or idle state. There is time delay between state conversions according to the IEC62386-102 standard specification.

In DALI standard definition, every configuration command shall be received twice within 100ms to reduce the probability of incorrect reception. If received correctly twice, the configuration command should be executed and then go to idle state. Otherwise it goes to idle state.

A backward frame shall only be sent after the reception of a query command or a write memory command. If backward frame is NO, the DALI slave should not react. Otherwise it should send the DALI answer signal to master. After this is done, it should go to idle mode with a delay.

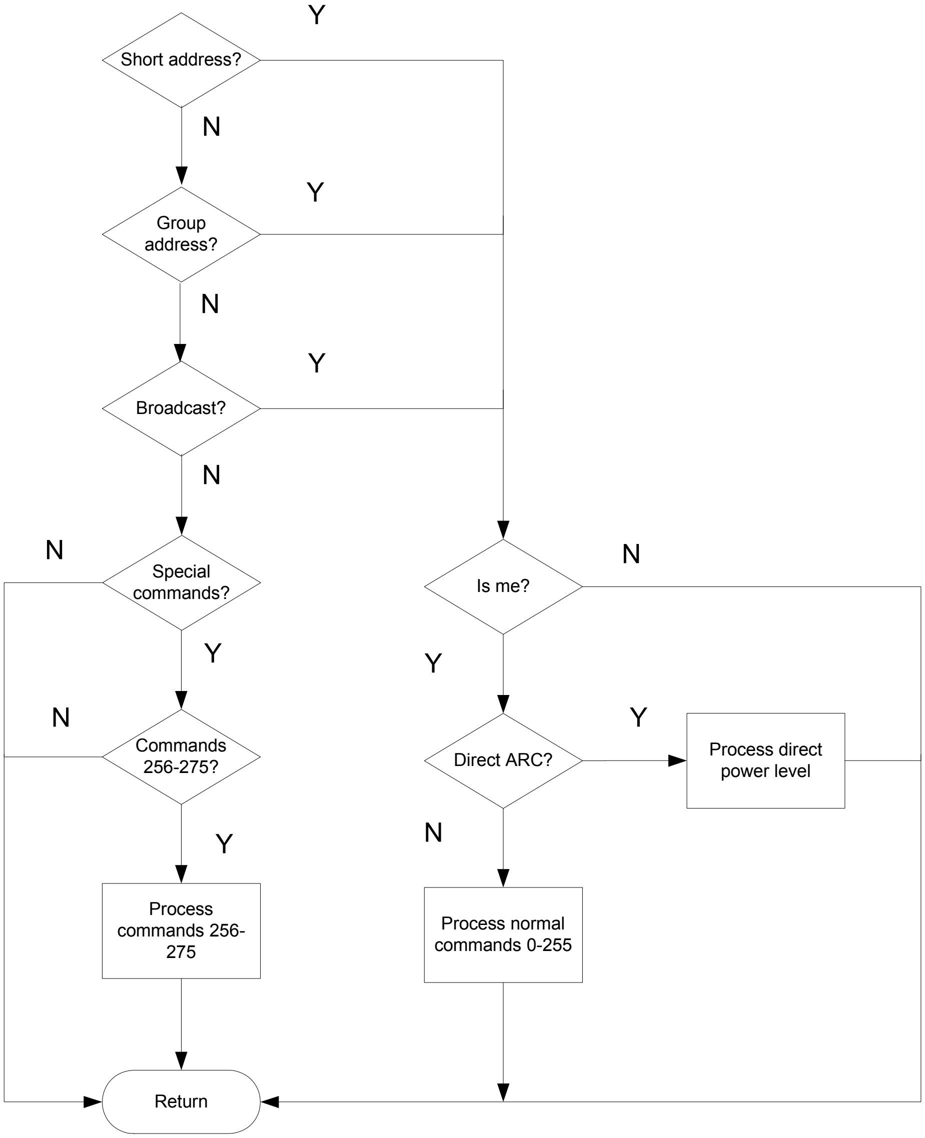
Figure 3-2. Frame State Convert diagram.



3.2.2 Frame message process

After DALI slave receives forward frame, it should decode the two bytes of address and data. The address type includes short address, group address and broadcast. Address and data compose the command code. For every command code, the corresponding defined function should be executed. In this module, the slave shall decode the address and call the command function.

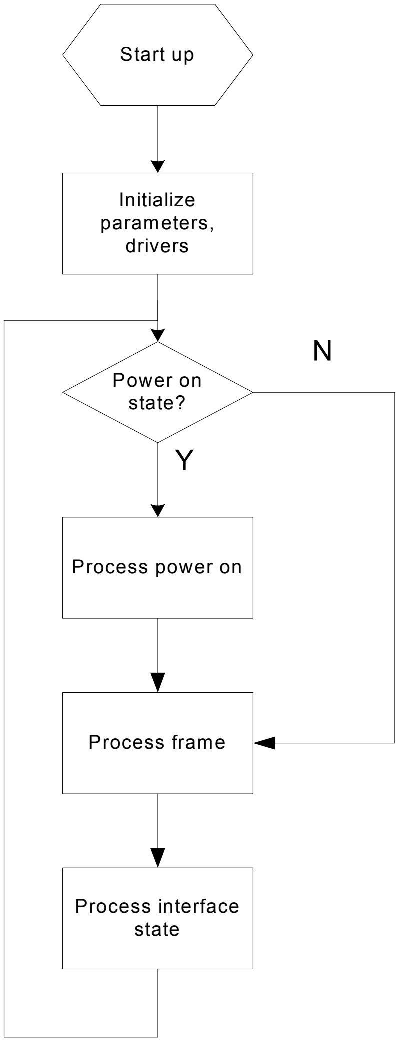
Figure 3-3. Message process flowchart.



3.3 Application

The whole system process is an infinite loop. After start up, it initializes the parameters and configures microcontroller blocks. In the first power on period, if no command affecting power level is received, it shall go to power on level immediately without fading. The system also processes frame, interface state detection sequentially.

Figure 3-4. Message process flowchart.



3.4 Logarithm dimming table

Here describes the dimming lookup table used in firmware.

3.4.1 high_PWM_val and low_PWM_val

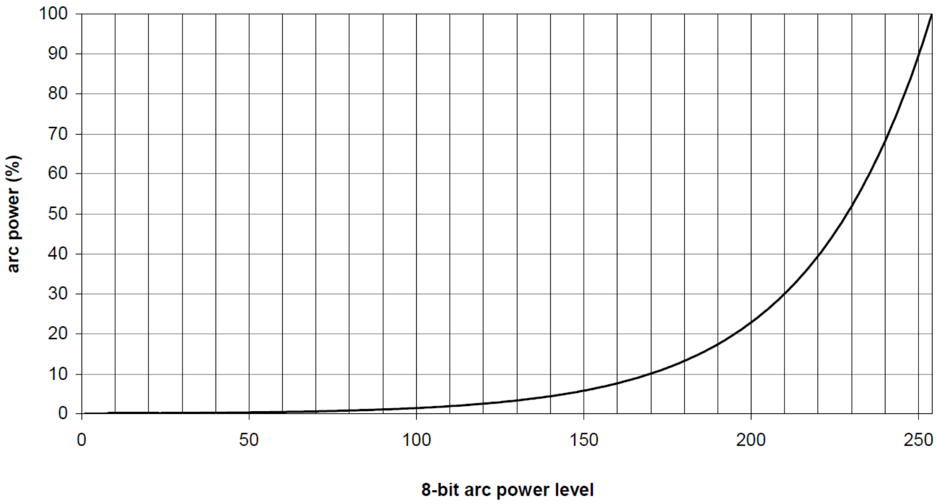
DALI standards define the logarithmic dimming curve between power value and power level. The formula below defines their relation.

$$X(n) = 10^{\frac{n-1}{253/3} - 1} \qquad \left| \frac{X(n) - X(n+1)}{X(n)} \right| = \text{constant} = 2.8\ \%$$

where X denotes power value and n denotes power level.

Power level ranges from 1 to 254, while corresponding power value ranges from 0.1% to 100%. Figure 3-5 shows their relationship.

Figure 3-5. Logarithm dimming curve.



In this example, 12-bit PWM of the MCU is used to output power. PWM provides a duty ratio from 0 to 4095 which corresponds to a precision of min. 0.1% power value. The PWM duty ratio lookup table is stored in arrays *high_PWM_val* and *low_PWM_val*. The tables are stored in EEPROM to save FLASH space.

Timer1 is used as the time base for 12-bit PWM. The initialization is done in function *dali_tc_init()*.

```
/*Timer1 clock with 8 prescaling */
tc_write_clock_source(TC1, TC_CLKSEL_DIV8_gc);
/* OC1B in Compare Match, PB2 output, non-inverting, Fast PWM,
 * ICR1 is top*/
TCCR1A = (0 << COM1A1) | (0 << COM1A0) | (1 << COM1B1) | (0 << COM1B0) | (1 << WGM11) | (0 << WGM10);
TCCR1B |= (1 << WGM13) | (1 << WGM12);
/* Used as top value for PWM, 0x0fff makes this a 12-bit PWM */
ICR1 = TC1_TOP;
/* Clear counter */
TCNT1 = 0;
```

3.4.2 flash_fade_rate_val and flash_inv_fadetime_val

DALI standard defines FADE TIME and FADE RATE as light dim parameters. FADE TIME is the time taken for the light to change from the current dim level to the target dim level. FADE RATE is the speed of the light change.

There are sixteen levels in the formula that:

- FADE TIME = $2^{(X/2-1)}$
- FADE RATE = 253 / (FADE TIME)

Their value relationship is given in Table 3-1.

Table 3-1. FADE TIME and FADE RATE.

X	FADE TIME [s]	FADE RATE [steps/s]
0	No fade	Not applicable
1	0.7	358
2	1.0	253
3	1.4	179

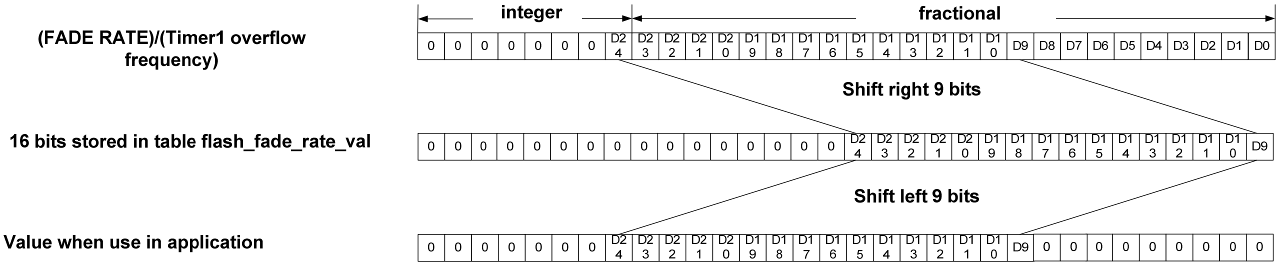
4	2.0	127
5	2.8	89.4
6	4.0	63.3
7	5.7	44.7
8	8.0	31.6
9	11.3	22.4
10	16.0	15.8
11	22.6	11.2
12	32.0	7.9
13	45.3	5.6
14	64.0	4.0
15	90.5	2.8

In this design, Timer1 is used as timer of FADE TIME and FADE RATE, and its overflow interrupt is used as time base for light dimming.

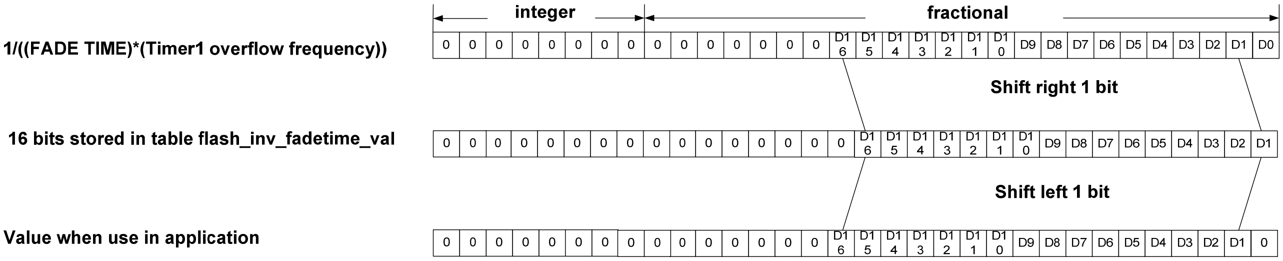
32-bit variable *dali.long_level* is used to store power level whose lower 24 bits is the fractional value and upper 8 bits is the actual DALI level.

```
union daliunion {
        //dali.long_level is used during fade - allows smoother fading between fixed DALI levels.
        uint32_t  long_level;
        struct {
                // Lower three bytes store the fractional value of DALI level
                uint8_t   byte0;
                uint8_t   byte1;
                uint8_t   byte2;
                // Upper 8 bits dali.level is the actual DALI level.
                uint8_t   level;
                };
}dali;
```

For DALI commands UP and DOWN, FADE RATE is used for light dimming. In every timer period, the change of DALI level is FADE RATE divided by Timer1 overflow frequency. Thus a 25-bits result is got with lower 24 bits used as fractional. To save space while keep precision, the result is right-shifted by 9 bits and the remainder 16-bits value is stored in lookup table *flash_fade_rate_val*. And accordingly in application, the value from lookup table should be left-shifted by 9 bits before being used.


integer
fractional
(FADE RATE)/(Timer1 overflow frequency)
0 0 0 0 0 0 0 D24 D23 D22 D21 D20 D19 D18 D17 D16 D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
Shift right 9 bits
16 bits stored in table flash_fade_rate_val
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D24 D23 D22 D21 D20 D19 D18 D17 D16 D15 D14 D13 D12 D11 D10 D9
Shift left 9 bits
Value when use in application
0 0 0 0 0 0 0 D24 D23 D22 D21 D20 D19 D18 D17 D16 D15 D14 D13 D12 D11 D10 D9 0 0 0 0 0 0 0 0 0

For DALI direct power control command, FADE TIME is used for light dimming. In every timer period, the change of DALI level is (1/FADE TIME) divided by Timer1 overflow frequency. Thus a 17-bits result is got. Also to save space, the result is right-shifted by 1 bit and the remainder 16-bits value is stored in lookup table *flash_inv_fadetime_val*. And accordingly in application, the value from lookup table should be left-shifted by 1 bit before being used. To dim *N* levels in power control command, the above result needs to be multiplied by *N*.



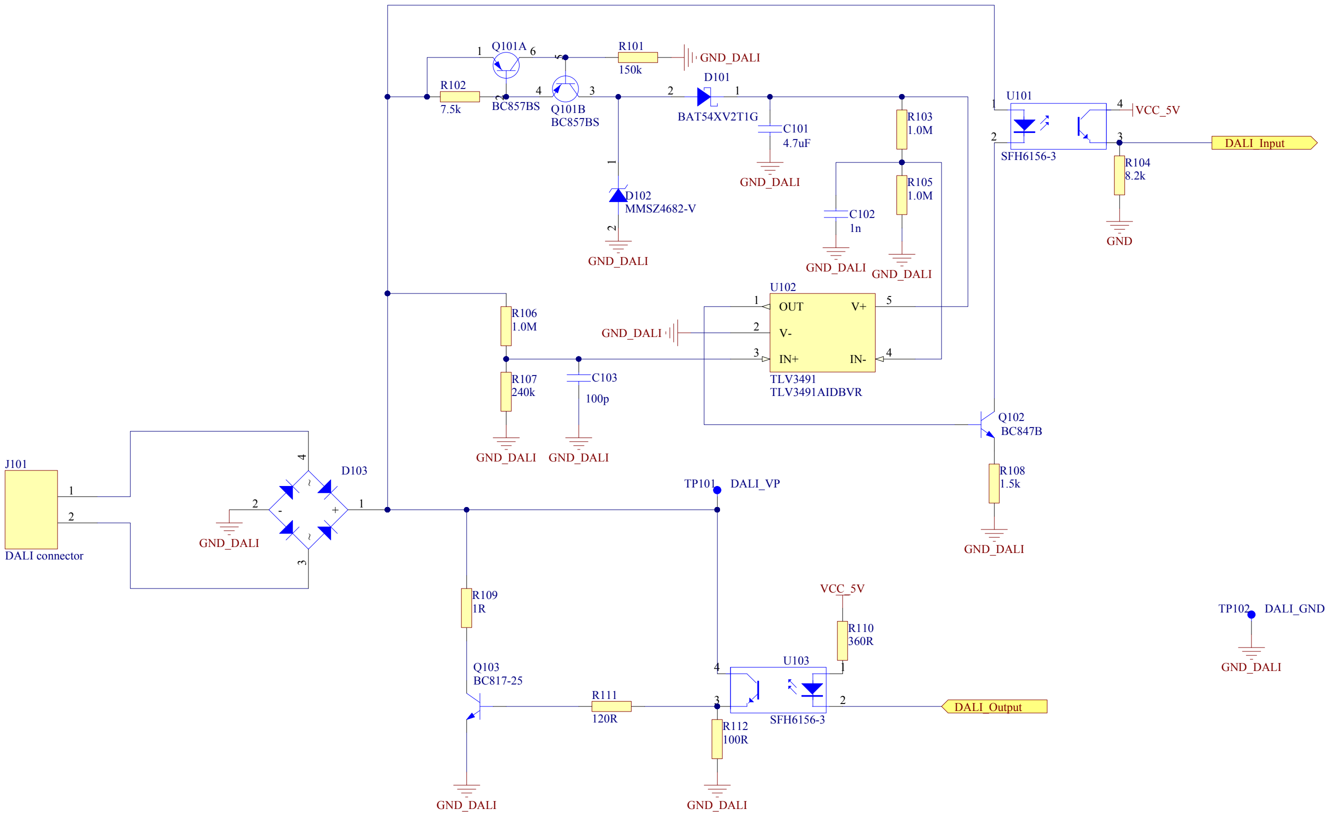
1/((FADE TIME)*(Timer1 overflow frequency))	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
16 bits stored in table flash_inv_fadetime_val	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1
Value when use in application	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	0

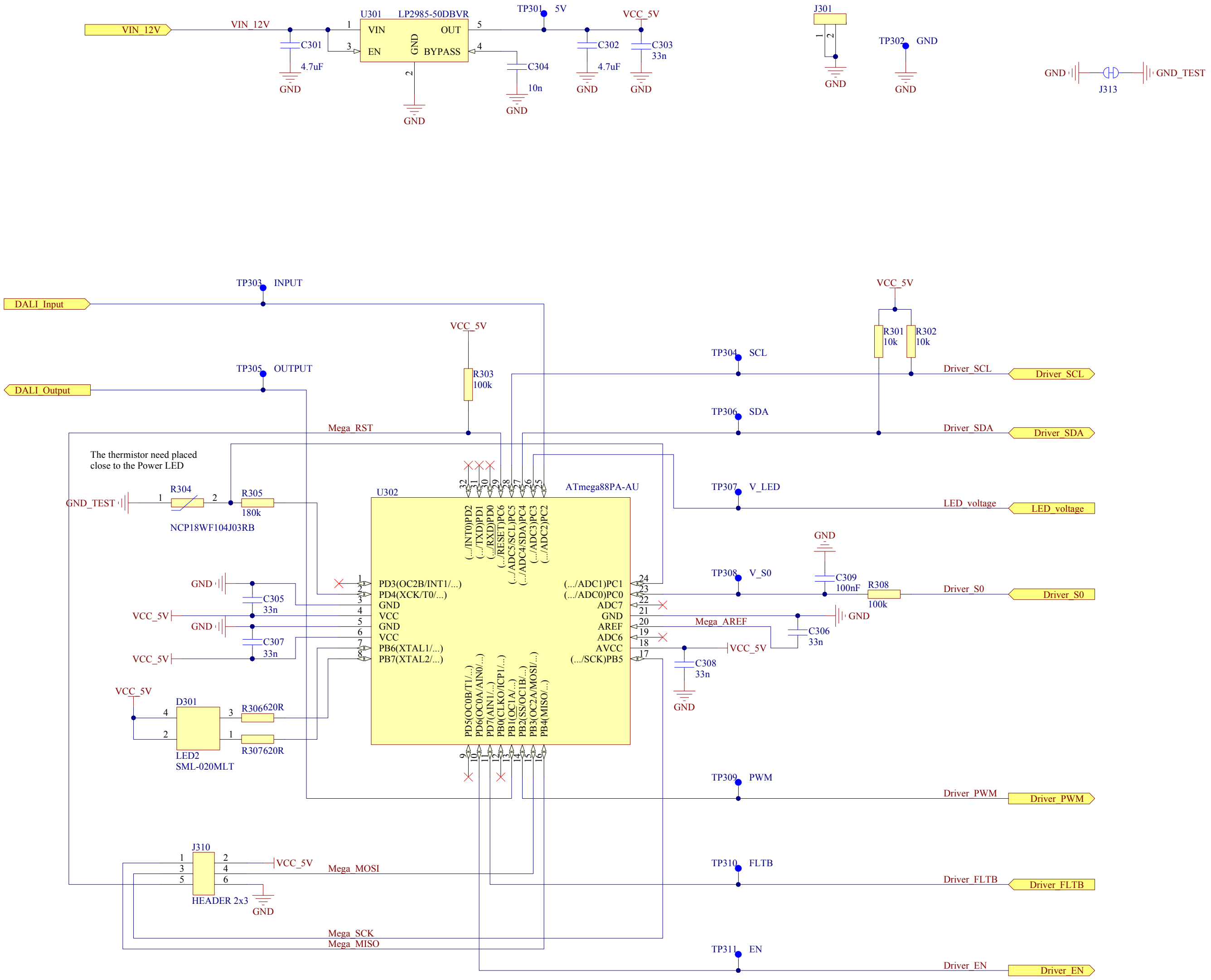
Appendix A. Command Sets

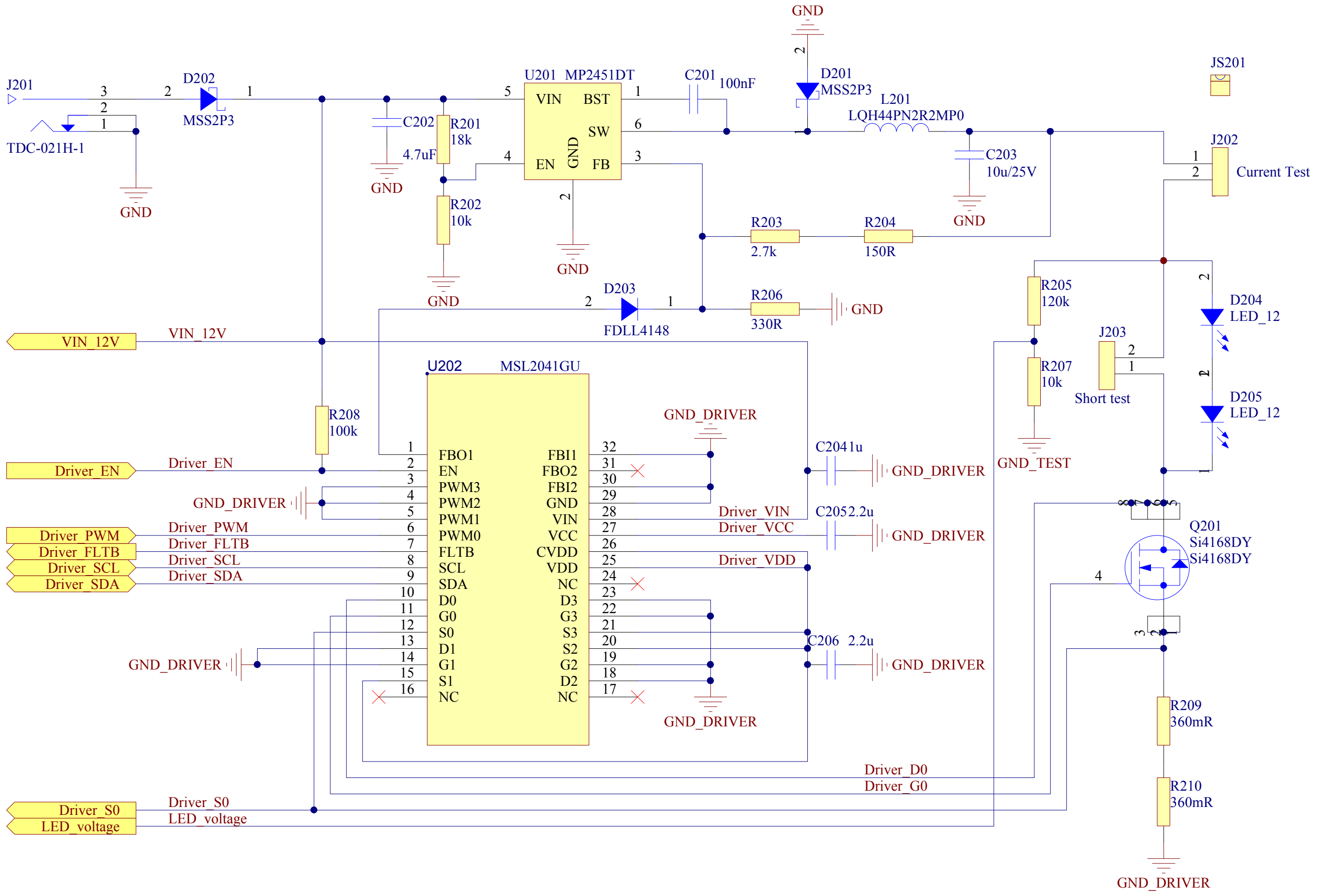
Command number	Command name
–	DIRECT ARC POWER CONTROL
0	OFF
1	UP
2	DOWN
3	STEP UP
4	STEP DOWN
5	RECALL MAX LEVEL
6	RECALL MIN LEVEL
7	STEP DOWN AND OFF
8	ON AND STEP UP
9	ENABLE DAPC SEQUENCE
10 – 15	Reserved
16 – 31	GO TO SCENE
32	RESET
33	STORE ACTUAL LEVEL IN THE DTR
34 – 41	Reserved
42	STORE THE DTR AS MAX LEVEL
43	STORE THE DTR AS MIN LEVEL
44	STORE THE DTR AS SYSTEM FAILURE LEVEL
45	STORE THE DTR AS POWER ON LEVEL
46	STORE THE DTR AS FADE TIME
47	STORE THE DTR AS FADE RATE
48 – 63	Reserved
64 – 79	STORE THE DTR AS SCENE
80 – 95	REMOVE FROM SCENE
96 – 111	ADD TO GROUP
112 – 127	REMOVE FROM GROUP
128	STORE DTR AS SHORT ADDRESS
129	ENABLE WRITE MEMORY
130 – 143	Reserved
144	QUERY STATUS
145	QUERY CONTROL GEAR
146	QUERY LAMP FAILURE
147	QUERY LAMP POWER ON
148	QUERY LIMIT ERROR
149	QUERY RESET STATE
150	QUERY MISSING SHORT ADDRESS
151	QUERY VERSION NUMBER
152	QUERY CONTENT DTR
153	QUERY DEVICE TYPE
154	QUERY PHYSICAL MINIMUM LEVEL
155	QUERY POWER FAILURE
156	QUERY CONTENT DTR1
157	QUERY CONTENT DTR2

158 – 159	Reserved
160	QUERY ACTUAL LEVEL
161	QUERY MAX LEVEL
162	QUERY MIN LEVEL
163	QUERY POWER ON LEVEL
164	QUERY SYSTEM FAILURE LEVEL
165	QUERY FADE TIME/FADE RATE
166 – 175	Reserved
176 – 191	QUERY SCENE LEVEL (SCENES 0-15)
192	QUERY GROUPS 0-7
193	QUERY GROUPS 8-15
194	QUERY RANDOM ADDRESS (H)
195	QUERY RANDOM ADDRESS (M)
196	QUERY RANDOM ADDRESS (L)
197	READ MEMORY LOCATION
198 – 223	Reserved
224 – 254	See parts 207 of this standard
255	QUERY EXTENDED VERSION NUMBER
256	TERMINATE
257	DATA TRANSFER REGISTER (DTR)
258	INITIALISE
259	RANDOMISE
260	COMPARE
261	WITHDRAW
262 – 263	Reserved
264	SEARCHADDRH
265	SEARCHADDRM
266	SEARCHADDRL
267	PROGRAM SHORT ADDRESS
268	VERIFY SHORT ADDRESS
269	QUERY SHORT ADDRESS
270	PHYSICAL SELECTION
271	Reserved
272	ENABLE DEVICE TYPE X
273	DATA TRANSFER REGISTER 1 (DTR1)
274	DATA TRANSFER REGISTER 2 (DTR2)
275	WRITE MEMORY LOCATION
276 – 349	Reserved

Appendix B. Schematic




VIN_12V
VIN_12V
C301
4.7uF
GND
U301
LP2985-50DBVR
1
VIN
3
EN
GND
2
OUT
5
BYPASS
4
GND
C304
10n
GND
TP301
5V
VCC_5V
C302
4.7uF
GND
C303
33n
GND
J301
1
2
GND
TP302
GND
GND
GND
J313
GND_TEST
TP303
INPUT
DALI_Input
TP305
OUTPUT
DALI_Output
VCC_5V
R303
100k
Mega_RST
The thermistor need placed
close to the Power LED
GND_TEST
R304
1
2
NCP18WF104J03RB
R305
180k
U302
ATmega88PA-AU
32
31
30
29
28
27
26
25
(.../INT0)PD2
(.../TXD)PD1
(.../RXD)PD0
(.../RESET)PC6
(.../ADC5/SCL)PC5
(.../ADC4/SDA)PC4
(.../ADC3)PC3
(.../ADC2)PC2
1
2
3
4
5
6
7
8
PD3(OC2B/INT1/...)
PD4(XCK/T0/...)
GND
VCC
GND
VCC
PB6(XTAL1/...)
PB7(XTAL2/...)
24
23
22
21
20
19
18
17
(.../ADC1)PC1
(.../ADC0)PC0
ADC7
GND
AREF
ADC6
AVCC
(.../SCK)PB5
9
10
11
12
13
14
15
16
PD5(OC0B/T1/...)
PD6(OC0A/AIN0/...)
PD7(AIN1/...)
PB0(CLKO/ICP1/...)
PB1(OC1A/...)
PB2(SS/OC1B/...)
PB3(OC2A/MOSI/...)
PB4(MISO/...)
GND
C305
33n
VCC_5V
GND
C307
33n
VCC_5V
VCC_5V
D301
4
2
3
1
LED2
SML-020MLT
R306620R
R307620R
VCC_5V
R301
10k
R302
10k
TP304
SCL
Driver_SCL
Driver_SCL
TP306
SDA
Driver_SDA
Driver_SDA
TP307
V_LED
LED_voltage
LED_voltage
GND
TP308
V_S0
C309
100nF
R308
100k
Driver_S0
Driver_S0
GND
Mega_AREF
C306
33n
VCC_5V
C308
33n
GND
TP309
PWM
Driver_PWM
Driver_PWM
J310
1
3
5
2
4
6
VCC_5V
Mega_MOSI
HEADER 2x3
GND
TP310
FLTB
Driver_FLTB
Driver_FLTB
Mega_SCK
Mega_MISO
TP311
EN
Driver_EN
Driver_EN


J201
TDC-021H-1
D202
MSS2P3
GND
VIN_12V
Driver_EN
Driver_PWM
Driver_FLTB
Driver_SCL
Driver_SDA
Driver_S0
LED_voltage
C202
4.7uF
R201
18k
R202
10k
U201 MP2451DT
VIN
BST
SW
EN
GND
FB
C201
100nF
D201
MSS2P3
L201
LQH44PN2R2MP0
C203
10u/25V
R203
2.7k
R204
150R
D203
FDLL4148
R206
330R
R208
100k
U202
MSL2041GU
FBO1
EN
PWM3
PWM2
PWM1
PWM0
FLTB
SCL
SDA
D0
G0
S0
D1
G1
S1
NC
FBI1
FBO2
FBI2
GND
VIN
VCC
CVDD
VDD
NC
D3
G3
S3
S2
G2
D2
NC
GND_DRIVER
Driver_VIN
Driver_VCC
Driver_VDD
Driver_D0
Driver_G0
C2041u
C2052.2u
C206 2.2u
JS201
J202
Current Test
R205
120k
R207
10k
J203
Short test
GND_TEST
D204
LED_12
D205
LED_12
Q201
Si4168DY
Si4168DY
R209
360mR
R210
360mR

Appendix C. Revision History

Doc. Rev.	Date	Comments
42071B	07/2013	Add hardware information in “Description” chapter
42071A	02/2013	Initial document release

Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Building
1-6-4 Osaki
Shinagawa-ku, Tokyo 141-0032
JAPAN
Tel: (+81)(3) 6417-0300
Fax: (+81)(3) 6417-0370

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.